CDF/DOC/ELECTRON/PUB/xxxx
April 15, 1999
Version 1.0

CDF SHOWER MAX
SMD DIAGNOSTICS INTERFACE

Craig Drennan
Fermi National Accelerator Laboratory
PO Box 500, MS222, Batavia, IL 60510
(630) 840-2160, cdrennan@fnal.gov

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. Introduction

The SMD Crate Diagnostics Interface (SMDI) provides a means for testing and diagnostic programs running on a PC computer to monitor and control signals in the Shower Max Digitizer Crate [1]. Through the SMDI the PC can perform the following functions.

1. Emulate the SMXR Shower Max Readout Board ability to send SMQIE control signals including the system clock, and receive data transmitted from a single SMD Digitizer Board.
2. Place the SMD electronics into a single channel diagnostics mode and monitor the data transmission from a selected SMQIE channel in the crate.
3. Monitor the operation of the SMC Crate Controller by monitoring a selected set of the sequencing and control signals generated on the crate controller.

The Diagnostic Interface is connected to the PC through either an enhanced parallel port interface using the ECP protocol or through the CDF Particle Instrumentation Group's ISA bus interface card. Both of these interfaces will allow the code running on the PC to Read and Write all of the I/O registers on the SMDI. The PC Windows software currently written to make use of the SMDI uses a Universal Serial Bus (USB) to enhanced parallel port cable for which drivers were available. A separate document describes the interface software and user interface, [2].

The SMDI connects to the SMD Crate in three ways. Control signals are transmitted to the SMC Crate Controller over a thirty-four pair cable connected to the port normally used for the SMXR to SMC control link. Data from the SMQIE's on the SMD Digitizer Boards are received over a thirteen pair cable connected to the data output port of a selected SMD board in the crate. The thirteen pair cable can also be connected to a second port on the SMC Crate Controller which is used specifically to transmit the sequence and control signals generated by the SMC to the diagnostics for monitoring proposes.

The SMDI uses an eight bit Transmit FIFO clocked at 33 ns to produce sequences for the eight control signals which have system timing requirements. That is the bits of each byte loaded into the Transmit FIFO represent the level of the signal associated with each bit during a certain 33 ns interval in time.

The SMDI uses two eight bit Receive FIFO's to receive the data transmitted to the interface from either the SMD boards or the SMC crate controller. In the case of the SMD data, data is clocked into the Receive FIFO by the 33 ns data strobe from the SMD boards. In the case of the data from the SMC, data is clocked in on the falling edge of the 132 ns system clock transmitted from the SMC.

This document will focus on how the Diagnostics interface generates the signals and handles the necessary timing, and interfaces with the PC. A more detailed description

of the signals in the SMD crate is provided in the design note “CDF Shower Max SMC Crate Controller”, [1].

II. Control Signal Generation and Transmission

II.1 The Clocks

Three clocks are generated on the Diagnostics Interface. A programmable clock oscillator (ICD6233, by IC Designs) generates a 132 ns clock which is used as the reference clock of a Motorola MC88915 low skew CMOS PLL clock driver. From this PLL driver we take a 132 ns clock, a 66 ns clock and a 33 ns clock.

The 132 ns clock is transmitted from the SMDI as the SMD crate system clock that is used to clock the SMQIE’s and time the data transmission sequencing signals and data strobe signals. The 132 ns clock is also used to clock the Capacitor Phase counter in the programmable logic.

The 66 ns clock is used to clock the logic on the SMDI that generates signals used on the SMDI for such things as resetting the FIFO’s and timing the PC Read and Write signals. Table II.1 lists the signals that are timed with the 66 ns clock.

The 33 ns clock is used to clock data out of the Transmit FIFO producing sequences for the eight control signals that have a predetermined timing relationship to one another. Each bit of each byte loaded into the Transmit FIFO represent the level of the signal associated with the bit during a certain 33 ns interval in time.

Table II.1.1 Signals timed with the 66 ns clock.

Signal	Signal Description
Read Port Select Signals from the PC	Logic in PLD1, U3, uses the clock to guarantee that the output enable (chip selects) generated in response to a PC Read are active (low) for 66 ns beyond the rising edge of the Read signals. This is done to provide plenty of data hold time.
Write Port Select Signals from the PC	Logic in PLD1, U3, uses the clock to generate data Write signals whose rising edge occurs 66 ns after the falling edge of the specific Write Port Select signal from the PC. This provides a 66 ns setup time for the registers and ensures that the Write pulse rising edge will clock in data from the PC while it is still valid.
Receiver FIFO Reset	Logic in PLD2, U14, uses the clock to generate a 66 ns to 132 ns pulse to reset the receiver FIFO’s when a High is written to bit 0 of Register 3.
Receiver FIFO Retransmit	Logic in PLD2, U14, used the clock to generate a 66 ns to 132 ns pulse to signal a retransmission from the receiver FIFO’s when a High is written to bit 1 of Register 3.
Transmit FIFO Reset	Logic in PLD2, U14, uses the clock to generate a 66 ns to 132 ns pulse to reset the Transmit FIFO when a High is written to bit 2 of Register 3.

Transmit FIFO Retransmit	Logic in PLD2, U14, used the clock to generate a 66 ns to 132 ns pulse to signal a retransmission from the transmit FIFO when a High is written to bit 3 of Register 3.
Start Receiver Command	Logic in PLD2, U14, uses the clock to generate a 66 ns to 132 ns pulse to start reception of data into the receiver FIFO's (in certain receiver modes) when a High is written to bit 5 of Register 3.
Stop Receiver Command	Logic in PLD2, U14, uses the clock to generate a 66 ns to 132 ns pulse to stop reception of data into the receiver FIFO's (in certain receiver modes) when a High is written to bit 6 of Register 3.
Start Transmission Command	Logic in PLD2, U14, uses the clock to generate a 66 ns to 132 ns pulse to start transmission of data from the transmit FIFO when a High is written to bit 7 of Register 3.
Receive FIFO Read Clocks	Logic in PLD1, U3, uses the clock to generate a Read clock which has rising edge 66 ns after the falling edge of the receiver Read select signal. The FIFO's are of the synchronous variety and need a Read clock as well as a chip select.
Load DAC Strobe	Logic in PLD2, U14, uses the clock to generate a 198 ns active low pulse used by the SMC crate controller to load data into the DAC of the calibration voltage reference. This occurs when a High is written to bit 4 of Register 3.
Optional Clock for Counter 2	This clock can be used by Counter 2 which can be used to time the occurrence of test stimulus into the SMQIE's.

II.2 Timed Signals from the Transmit FIFO

The signals that are generated by the Transmit FIFO are listed in Table II.2.1. A typical sequence of events involving these signals would be as follows.

1. The SMQIE's would be Reset. Capacitor Phase counters in the diagnostic interface and the SMQIE's are reset and then advance with each rising edge of the 132ns clock.
2. The external input stimulus for the SMQIE's would be fired and forty-five clocks after this the digitized results of this input emerges from the QIE pipeline.
3. At this time a Level 1 Accept would be indicated and the data from the pipeline stored in one of four buffers indicated by the value of the bits SMWBUFF[1:0].
4. After this a Data Transmit Request is made and the stored data is transmitted from the SMD boards from one of the four SMQIE buffers indicated by SMRBUFF[1:0].

Table II.2.1 Signals Timed Using the Transmit FIFO

Signal	FIFO Bit	Description
TRIG_OUT	Bit 0	Triggers the external stimulus to the SMQIE's
SMRESET	Bit 1	SMD Crate Reset used to initialize the SMQIE's and other logic in the crate. Note that SMRESET can also be generated by Writing a High to bit 4 of Register 1. The later method is necessary to get the cap phase counter on the SMDI in sync with the SMQIE cap phase. It is the cap phase counter on the SMDI that determines on which clock the sequence from the Transmit fifo begins.
SML1A	Bit 2	The Level 1 Accept signal that is used in conjunction with the Write Buffer bits to direct the SMQIE's to store the data word emerging from the QIE pipeline in the buffer specified by the Write Buffer bits for later transmission to the SMXR readout board.
SMDTR	Bit 3	The Data Transmit Request signal that is used in conjunction with the Read Buffer bits to direct the SMQIE's to transmit the data stored in the buffer specified by the Read Buffer bits.
SMWBUFF1	Bit 4	Write buffer bit 1
SMWBUFF0	Bit 5	Write buffer bit 0
SMRBUFF1	Bit 6	Read buffer bit 1
SMRBUFF1	Bit 7	Read buffer bit 1

Data from the Transmit FIFO is clocked out by a 33 ns clock that is synchronized to the 132 ns clock that is transmitted from the SMDI to the SMD crate. This allows us to change each of the eight signal generated by the FIFO to change four times per 132 ns clock or otherwise have pulse widths as small as 33 ns.

There are some special timing requirements for the Level 1 Accept and the Write buffer select signals with respect to the 132ns clock (as measured at the SMQIE's). The requirements are determined by the SMQIE design and are presented in Figure II.2.1 and Table II.2.2. These timing issues are managed by the SMC Crate Controller, [1].

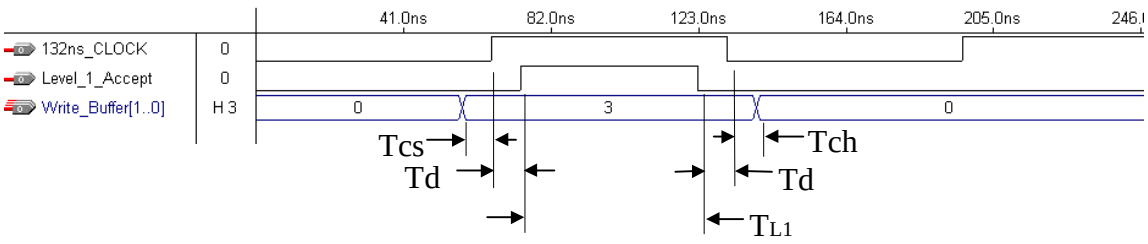

Figure II.2.1 Timing for the Level 1 Accept.

Table II.2.2 Timing parameters for the Level 1 Accept.

Symbol	Parameter	Minimum
Tcs	Clock setup time before Level 1 Accept goes high	> 0 ns
Tch	Clock hold time after Level 1 Accept goes low	> 0 ns
Tds	Write Buffer Select setup time before Clock goes high	5 ns
Tdh	Write Buffer Select hold time after Clock goes low	5 ns
TL1	Level 1 Accept pulse width	20 ns

II.3 Counter Outputs

There are two 11 bit Counter/Timers on the Diagnostics Interface. The first is the Receiver Counter which, in certain modes, enables and disables the receiver. The second is the Auxiliary Counter. Each is a count down timer that can be loaded with an initial count value and a count match value. When the output of a counter is equal to or less than its match value the counter MATCH signal is High. When the output of a counter reaches zero the counter stops and the counter CNT_STOPPED signal goes High.

The MATCH and CNT_STOPPED signals of the Auxiliary Counter are tied into the programmable logic device PLD2, U14. Within PLD2, U14, these signals are used to generate the AUX_TRIGGER and AUX_OUTPUT signals. There is some small amount of flexibility in the logic that generates the signals.

The AUX_TRIGGER signal is OR'd with the TRIG_OUT signal from the Transmit FIFO. Both signals are driven by the SMDI to trigger an external stimulus input for the SMQIE's. The AUX_OUTPUT is an extra output. Both signals are driven out of the SMDI box with LVDS drivers.

The Auxiliary Counter can be clocked with the 66 ns clock or with the 66 ns clock divided down to a 132 ns clock. The Receiver Counter is clocked with either the data strobe signal sent with the received data or with the 66 ns clock, depending on the receiver mode of operation.

II.4 Asynchronous Signals

There are several control signals transmitted from the SMDI that are not necessarily synchronized to one of the clocks or other signals. These signals are listed in Table II.3.1. These signals are control by the PC by writing certain registers in the SMDI.

Table II.4.1 Asynchronous control signals.

Signal	Description
MODESEL0 MODESEL1	The mode select signals are decoded on the SMC Crate Controller to set the operation of the crate into the following configurations. MODESEL[1:0] = 0 -- Normal DAQ Mode. MODESEL[1:0] = 1 -- Current Injection Enabled (Normal DAQ Mode). MODESEL[1:0] = 2 -- Source Monitoring Enabled (Normal DAQ Mode). MODESEL[1:0] = 3 -- PATTEN Diagnostic. These signals are controlled by writing to bit 1 and bit 2 of Register 1.
SML2R	The SML2 is the Level 2 Reject signal which halts transmission from the SMQIE's in the normal mode. SML2R is controlled by writing to bit 3 of Register 1
DIAG_DAC_STRB	This is the DAC strobe signal used to load the calibration voltage reference on the SMC Crate Controller. This signal is triggered by writing a High to bit 4 of Register 3.
ALT_SMRESET	This is signal is OR'd with the SMRESET reset signal that is generated by the Transmit FIFO. This signal is controller by writing to bit 4 of Register 1.

III. Receiving Data

The Diagnostics interface uses a Receive FIFO into which data is clocked from one of two sources. The first is the data output port of the SMD Digitizer Boards. The second is the signal monitor port on the SMC Crate Controller. Data from the SMD boards are clocked into the Receiver FIFO with the 33 ns data strobe clock that is transmitted with the data. Data from the SMC board are clocked in on the falling edge of the 132 ns system clock that is transmitted with the data from the SMC.

Besides having a write clock input, the Receive FIFO has a write enable input which can be used to control when data is clocked into the receiver. The logic for the receiver on the SMDI is designed to control the receiver in four different receiver modes. These modes determine how data reception is started and how it is stopped. The receiver modes are selected by writing bit 6 and bit 7 in Register 1, RCV_MODE[0] and RCV_MODE[1], respectively.

III.1 Receiver Mode 0, [Disabled]

In Receiver Mode 0 the receiver is disabled.

III.2 Receiver Mode 1, [Manual Start, Manual Stop]

Receiver Mode 1 is only slightly more interesting than Mode 0. The receiver is enabled to receive data by writing a High to bit 5 of Register 3. The receiver is disabled by writing a High to bit 6 of Register 3. This mode is used in the SMXR emulation mode where data transmissions from the SMD Digitizer Boards are controlled by the Data Transmit Request signal from the SMDI Transmit FIFO. For each Data Transmit Request an SMD board will transmit data from the Read Buffer for each channel on the board once. The data strobe from the SMD boards, which clocks the receive FIFO's does not transition when data is not being transmitted.

III.3 Receiver Mode 2, [Manual Start, Auto-Counter Stop]

In Receiver Mode 2 the user manually enables the receiver with a command from the PC, but reception is halted automatically once a predetermined number of data words have been received. Connecting the SMD Data Strobe to the clock input of the Receiver Counter and preloading the counter with the number of data words we wish to receive does this. Once the counter has counted down to zero the CNT1_STOPPED signal disables the Receiver FIFO. This mode is used when operating the SMD crate in the bypass diagnostics mode with a constant charge current input to the SMQIE's. In this situation, data is being transmitted from the SMD board continuously.

III.4 Receiver Mode 3, [Auto-Timer Start, Auto-Timer Stop]

In Receiver Mode 3 the Receiver Counter is preloaded with an initial count and the match register is setup with a value that when reached will cause the receiver to be enabled. Once the Receiver Counter has reached zero the counter stops and the receiver is disabled. In this mode the Receiver Counter is clock with the 66 ns clock.

Also, in this mode, the Auxiliary Counter is used to trigger the external stimulus input to the SMQIE's. Both counters are clock with the 66 ns clock and are gated on at the same time. In this manner one can define a time window in which to received data has a fixed offset in time from the point that the external stimulus input to the SMQIE's is triggered.

This mode is used in the bypass diagnostics mode. By using 11 bit counters clocked at 66 ns, there are 135 microseconds in which to set up the sequence of triggering the stimulus and receiving data. If a new data point from the SMQIE's is transmitted every 132 ns and it is desired to receive 20 data points there are still better than 130 micro seconds in which to adjust the differential between the trigger and the start of the receiver. Or rather, if the differential between trigger and reception was less than 100 microseconds at least 220 consecutive data points could be recorded in the FIFO's.

IV. The PC Interface

IV.1 The Enhanced Parallel Port Interface

There are 16 registers defined in the SMDI. The programmable logic of PLD1, U3, can be programmed to allow addressing of these registers, Reading and Writing, by a number of means. The current implementation is a IEEE Std 1284-1994 Bi-directional Parallel Peripheral Interface, [3], in which the ECP mode is implemented to do the data transfer. In addition to this there is logic to control three types of data transfers.

IV.1.1 Writing Registers

The interface allows you to Write an individual register. In some cases the written data is stored. In the case of Register 0 and Register 1 writing a High to a specific bit will result is a specific action such as resetting a FIFO, loading a counter, or starting or stopping the receiver.

IV.1.2 Reading All Registers

The way that the parallel interface host Reads data from the peripheral dictates that packets no smaller than 64 bytes be read at a time. Therefore, when it is desired to know what value is in a particular register, all the registers are returned in a 64 byte packet. The packet will contain four copies of all 16 registers.

In this case, when Register 4 and Register 5 (the receiver FIFO's) are Read the FIFO pointer is not advanced. The FIFO outputs are driven onto the databus to be Read, but the FIFO's Read clock is not toggled.

IV.1.3 Reading The Receiver FIFO's

The receiver FIFO's are read out 512 bytes at a time. The lower eight bits of each two-byte receiver word are Read from the first of the two receiver FIFO's then the upper eight bits are Read from the second. Then the next word in the two FIFO's is Read. There are 256, two byte words Read at a time.

IV.2 The "PIG Tail" ISA Bus Interface

The former CDF Particle Instrumentation Group has developed an ISA bus card that decodes I/O register Reads and Writes. It extends the eight Read register select signals and eight Write register select signals out of the PC on ribbon cables along with an eight bit Read databus and an eight bit Write databus. This interface has been used to interface DOS based programs written in BASIC with external devices requiring a simple PC interface.

IV.3 The Commercial Digital I/O Card Interface

The second PC interface available on the SMDI is provided to allow any digital I/O card with at least 24 programmable I/O connections to be used. Such digital I/O cards

are available from many manufacturers and can be bought for nearly any bus standard used commercially for data acquisition. These cards also come with software drivers that work with popular programming languages on the more popular operating systems.

IV.4 The Interface Logic Circuit

A block diagram of the interface circuit for the SMDI is shown in Figure IV.4.1. It is fairly general and the programmable logic can be setup to encode and/or decode a variety of addressing schemes. The current version implements the IEEE-1284 ECP parallel port protocol. The parallel port connection is made through J3. Other types of communications interfaces can be implemented by connecting 34 position ribbon cables to J1 and J2, or mounting a daughter board with additional electronics at this point. The tables below provide the pin out for J1, J2 and J3.

Table IV.4.1 Pin out for J1 (Part # 3M3431-5302)

1. Ground	2. PORT_D[0]
3. Ground	4. PORT_D[1]
5. Ground	6. PORT_D[2]
7. Ground	8. PORT_D[3]
9. Ground	10. PORT_D[4]
11. Ground	12. PORT_D[5]
13. Ground	14. PORT_D[6]
15. Ground	16. PORT_D[7]
17. no connection	18.
19. Ground	20. PORT_A[0]
21. Ground	22. PORT_A[1]
23. Ground	24. PORT_A[2]
25. Ground	26. PORT_A[3]
27. Ground	28. PORT_A[4]
29. Ground	30. PORT_A[5]
31. Ground	32. PORT_A[6]
33. Ground	34. PORT_A[7]

Table IV.4.2 Pin out for J2 (Part # 3M3431-5302)

1. Ground	2. PORT_C[0]
3. Ground	4. PORT_C[1]
5. Ground	6. PORT_C[2]
7. Ground	8. PORT_C[3]
9. Ground	10. PORT_C[4]
11. Ground	12. PORT_C[5]
13. Ground	14. PORT_C[6]
15. Ground	16. PORT_C[7]
17. no connection	18.
19. Ground	20. PORT_A[0]
21. Ground	22. PORT_A[1]
23. Ground	24. PORT_A[2]
25. Ground	26. PORT_A[3]
27. Ground	28. PORT_A[4]
29. Ground	30. PORT_A[5]
31. Ground	32. PORT_A[6]
33. Ground	34. PORT_A[7]

Table IV.4.3 Pin out for J3 (36 pin Centronics)

1. nStrobe (PORT_C[0])	2. Data(0) (PORT_A[0])
3. Data(1) (PORT_A[1])	4. Data(2) (PORT_A[2])
5. Data(3) (PORT_A[3])	6. Data(4) (PORT_A[4])
7. Data(5) (PORT_A[5])	8. Data(6) (PORT_A[6])
9. Data(7) (PORT_A[7])	10. nAck (PORT_D[0])
11. Busy (PORT_D[1])	12. Perror (PORT_D[2])
13. Select (PORT_D[3])	14. nAutoFd (PORT_C[1])
15. no connection	16. Ground
17. Shield Ground	18. +5V from Peripheral (Testpoint Only)
19. Ground	20. PORT_A[0]
21. Ground	22. PORT_A[1]
23. Ground	24. PORT_A[2]
25. Ground	26. PORT_A[3]

27. Ground	28. PORT_A[4]
29. Ground	30. PORT_A[5]
31. nInit (PORT_C[2])	32. nFault (Pulled High Always)
33. Ground	34. no connection
33. External 5V (Testpoint Only)	34. nSelectIn (PORT_C[3])

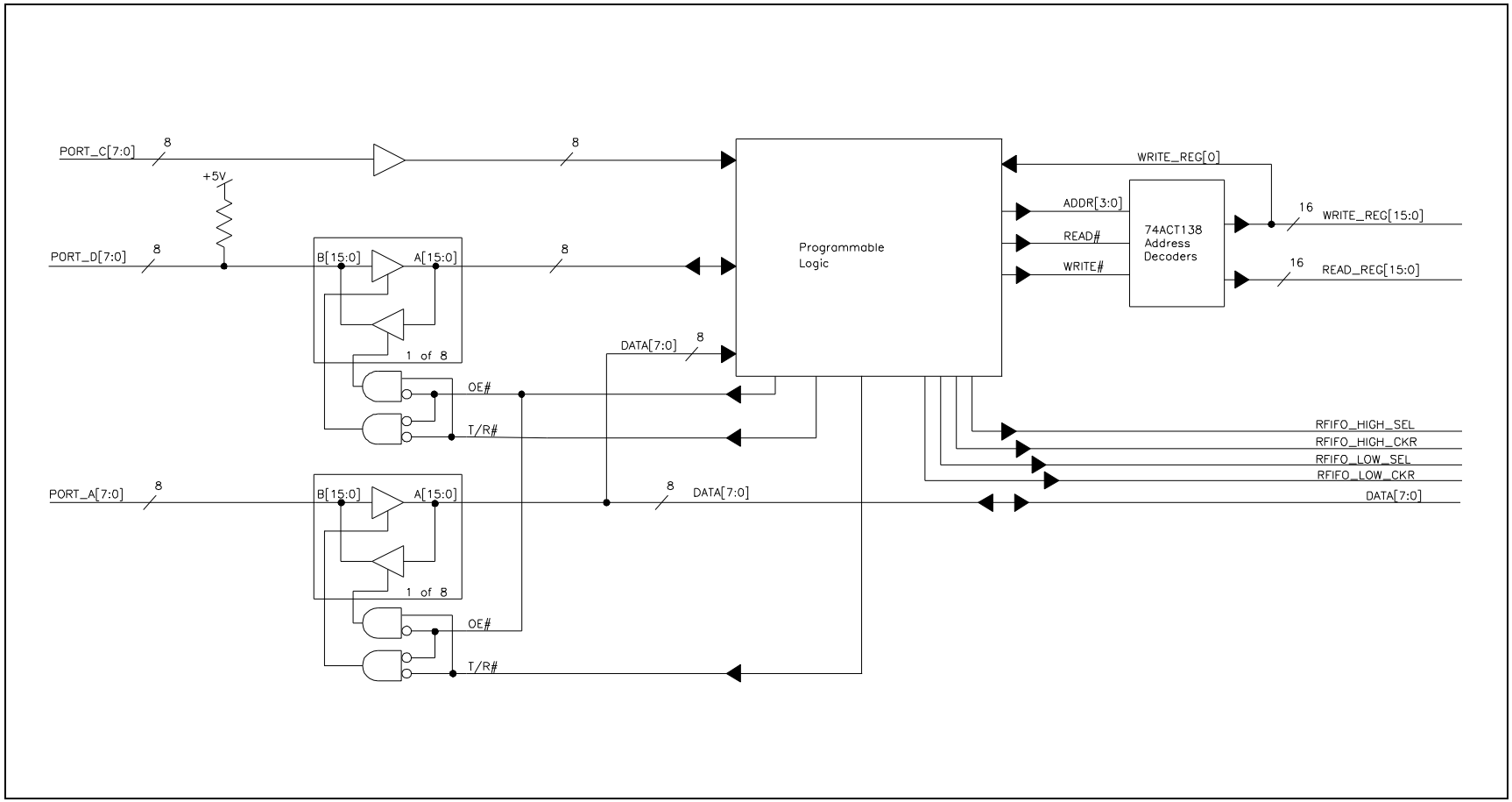

Figure IV.4.1 SMDI Host Interface Block Diagram

IV.5 Register Map for the Parallel Port Interface

The following tables describe the memory map for the SMDI register when using the parallel port interface. If another interface is to be used the logic in PLD1 will likely be different and map the registers in another manner.

Table IV.5.1 Register 0

Bit	Read Register 0	Write Register 0
0	--	--
1	--	--
2	--	Load Counters
3	--	Reset Counters
4	--	
5	--	
6	--	
7	--	

Table IV.5.2 Register 1

Bit	Read Register 1	Write Register 1
0	--	--
1	MODESEL0	MODESEL0
2	MODESEL1	MODESEL1
3	SML2R	SML2R
4	ALT_SMRESET	ALT_SMRESET
5	LOGIC RESET	LOGIC RESET
6	RCVR_MODE0	RCVR_MODE0
7	RCVR_MODE1	RCVR_MODE1

Table IV.5.3 Register 2

Bit	Read Register 2	Write Register 2
0:7	DIAGADDR[0:7]	DIAGADDR[0:7]

Table IV.5.4 Register 3

Bit	Read Register 3	Write Register 3 (Commands)
0	XFIFO_EMPTY (PLD 2 Output)	RFIFO_RESET
1	XFIFO_EF#	RFIFO_RETRANS
2	XFIFO_HF#	XFIFO_RESET
3	RFIFO_LOW_EF#	XFIFO_RETRANS
4	RFIFO_LOW_HF#	LOAD_DAC
5	RFIFO_HIGH_EF#	START_RCVR
6	RFIFO_HIGH_HF#	STOP_RCVR
7	RFIFO_WRITE_ENA#	XMT_FIFO_ENR

Table IV.5.5 Register 4

Bit	Read Register 4	Write Register 4
0:7	RFIFO_LOW_BYTE[0:7]	XFIFO[0:7]

Table IV.5.6 Register 5

Bit	Read Register 5	Write Register 5
0:7	RFIFO_HIGH_BYTE[0:7]	--

Table IV.5.7 Register 6

Bit	Read Register 6	Write Register 6
0:7	--	DAC_LOW_BYTE[0:7]

Table IV.5.8 Register 7

Bit	Read Register 7	Write Register 7
0:7	--	DAC_HIGH_BYTE[0:7]

Table IV.5.9 Register 8

Bit	Read Register 8	Write Register 8
0:7	RCV_COUNT_LOW[0:7]	RCV_COUNT_LOW[0:7]

Table IV.5.10 Register 9

Bit	Read Register 9	Write Register 9
0:7	RCV_COUNT_HIGH[0:7]	RCV_COUNT_HIGH[0:7]

Table IV.5.11 Register 10

Bit	Read Register 10	Write Register 10
0:7	RCV_MATCH_LOW[0:7]	RCV_MATCH_LOW[0:7]

Table IV.5.12 Register 11

Bit	Read Register 11	Write Register 11
0:7	RCV_MATCH_HIGH[0:7]	RCV_MATCH_HIGH[0:7]

Table IV.5.1 Register 12

Bit	Read Register 12	Write Register 12
0:7	AUX_COUNT_LOW[0:7]	AUX_COUNT_LOW[0:7]

Table IV.5.14 Register 13

Bit	Read Register 13	Write Register 13
0:7	AUX_COUNT_HIGH[0:7]	AUX_COUNT_HIGH[0:7]

Table IV.5.15 Register 14

Bit	Read Register 14	Write Register 14
0:7	AUX_MATCH_LOW[0:7]	AUX_MATCH_LOW[0:7]

Table IV.5.16 Register 15

Bit	Read Register 15	Write Register 15
0:7	AUX_MATCH_HIGH[0:7]	AUX_MATCH_HIGH[0:7]